

1 **Amendments to the Claims:**

2 Claims 1-11 and 13-36 were pending at the time the Office Action was
3 issued.

4 Claims 1, 8, 18, 23, 27, 32, and 33 are amended.

5 Claims 1-11 and 13-36 remain pending.

6
7 **1.** (Currently Amended) A method comprising:

8 receiving an original digital good;

9 randomly applying various forms of protection to a plurality of segments of
10 the original digital good to generate a plurality of protected segments to be
11 included in a protected digital good;

12 generating a plurality of checkpoints, each of the checkpoints being
13 associated with at least one of the protected segments, the checkpoint being
14 operable to cause a system receiving the protected digital good to invoke a
15 function call to validate that the at least one protected segment with which the
16 checkpoint is associated has not been tampered with based on at least one form of
17 protection applied to the at least one protected segment; and

18 assembling the protected digital good by collecting the plurality of
19 protected segments, wherein at least two of the segments overlap one another, the
20 overlapping segments being different from each other, and the checkpoints are
21 inserted in the protected digital good at varying positions outside of and relative to
22 the protected segments with which the checkpoints are associated.

1 2. (Original) A method as recited in claim 1, wherein the randomly applying
2 comprises pseudo randomly applying the various forms of protection according to
3 pseudo random techniques.

4
5 3. (Original) A method as recited in claim 1, wherein the applying comprises
6 randomly selecting the forms of protection from a set of available forms of
7 protection.

8
9 4. (Original) A method as recited in claim 1, wherein the applying comprises
10 applying the various forms of protection to randomly selected portions of the
11 original digital good.

12
13 5. (Original) A method as recited in claim 1, wherein the various forms of
14 protection are selected from a group of protection tools comprising code integrity
15 verification, acyclic code integrity verification, cyclic code integrity verification,
16 secret key scattering, obfuscated function execution, encryption/decryption,
17 probabilistic checking, Boolean check obfuscation, in-lining, reseeding pseudo
18 random number generators with time varying inputs, anti-disassembly methods,
19 varying execution paths between runs, anti-debugging methods, and time/space
20 separation between tamper detection and response.

21
22 6. (Original) A method as recited in claim 1, wherein the applying comprises
23 applying a form of protection in which a checksum can be computed on a set of
24 bytes of the digital good without actually reading the bytes.

1 7. (Original) A computer-readable medium comprising computer-readable
2 instructions that, when executed by a processor, direct a computer system to
3 perform the method as recited in claim 1.

4
5 8. (Currently Amended) A method comprising:
6 segmenting a digital good into a plurality of segments;
7 selecting multiple segments from the plurality of segments;
8 transforming only the selected segments according to different protection
9 techniques to produce a protected digital good having a composite of variously
10 protected segments;
11 augmenting at least one segment using a certain protection technique; and
12 inserting a checkpoint within the protected digital good but outside of the
13 augmented segment and at a varying position relative to the augmented segment,
14 the checkpoint being configured upon being encountered in the digital good to
15 invoke a function call suitable to evaluate a validity of to validate that the
16 augmented segment has not been tampered with based on the certain protection
17 techniques used to produce the at least one protected segment.

18
19 9. (Original) A method as recited in claim 8, wherein at least two of the
20 segments overlap one another.

21
22 10. (Original) A method as recited in claim 8, wherein the selecting comprises
23 randomly selecting the segments.
24
25

1 **11.** (Original) A method as recited in claim 8, wherein the transforming
2 comprises transforming the selected segments according to randomly chosen
3 protection techniques.

4
5 **12.** (Canceled).

6
7 **13.** (Original) A method as recited in claim 8, further comprising receiving
8 quantitative parameters indicative of how much the protected digital good should
9 be altered.

10
11 **14.** (Original) A method as recited in claim 13, wherein the transforming is
12 performed to satisfy the quantitative parameters.

13
14 **15.** (Original) A method as recited in claim 8, wherein the protection
15 techniques are selected from a group of protection tools comprising code integrity
16 verification, acyclic code integrity verification, cyclic code integrity verification,
17 secret key scattering, obfuscated function execution, encryption/decryption,
18 probabilistic checking, Boolean check obfuscation, in-lining, reseeding pseudo
19 random number generators with time varying inputs, anti-disassembly methods,
20 varying execution paths between runs, anti-debugging methods, and time/space
21 separation between tamper detection and response.

22
23 **16.** (Original) A method as recited in claim 8, wherein the transforming
24 comprises applying a protection technique in which a checksum can be computed
25 on a set of bytes of the digital good without actually reading the bytes.

1 17. (Original) A computer-readable medium comprising computer-readable
2 instructions that, when executed by a processor, direct a computer system to
3 perform the method as recited in claim 8.

4
5 18. (Currently Amended) A method comprising:

6 establishing parameters prescribing a desired quantity of protection to be
7 applied to a software product in generating a protected software product;

8 parsing the software product into code sections;

9 selecting at least one code section;

10 augmenting the selected code section to add protection qualities to generate
11 an augmented code section;

12 generating a checkpoint configured to cause a system receiving the
13 augmented code section to invoke a function call ~~attempt~~ to validate that the
14 augmented code section has not been tampered with based on the protection
15 qualities added to generate the augmented code section;

16 determining a checkpoint position for the checkpoint to be inserted in the
17 protected software product, the checkpoint position being outside of a position of
18 the augmented code section and at an offset to the augmented section that is varied
19 from additional checkpoint positions associated with other augmented code
20 sections; and

21 repeating the selecting and the augmenting for different code sections until
22 the desired quantity of protection has been applied.

23
24 19. (Original) A method as recited in claim 18, wherein the establishing
25 comprises enabling a user to enter the parameters.

1 **20.** (Original) A method as recited in claim 18, wherein the augmenting
2 comprises applying a protection technique selected from a group of protection
3 techniques comprising code integrity verification, acyclic code integrity
4 verification, cyclic code integrity verification, secret key scattering, obfuscated
5 function execution, encryption/decryption, probabilistic checking, Boolean check
6 obfuscation, in-lining, reseeding pseudo random number generators with time
7 varying inputs, anti-disassembly methods, varying execution paths between runs,
8 anti-debugging methods, and time/space separation between tamper detection and
9 response.

10
11 **21.** (Original) A method as recited in claim 18, wherein the augmenting
12 comprises applying a protection technique in which a checksum can be computed
13 on a set of bytes of the digital good without actually reading the bytes.

14
15 **22.** (Original) A computer-readable medium comprising computer-readable
16 instructions that, when executed by a processor, direct a computer system to
17 perform the method as recited in claim 18.

1 23. (Currently Amended) A production system, comprising:

2 a memory to store an original digital good;

3 a production server equipped with a set of multiple protection tools that
4 may be used to augment the original digital good for protection purposes, the
5 production server being configured to:

6 parse the original digital good and apply protection tools selected
7 from the set of protection tools only to selected portions of the original digital
8 good in a random manner to produce a protected digital good having a composite
9 of the protected selected portions;

10 generate a plurality of checkpoints, each of the checkpoints being
11 associated with and positioned outside of one of the protected selected portions
12 and causing a system receiving the protected digital good, upon encountering each
13 of the checkpoints, to invoke a function call ~~to attempt to~~ validate the protected
14 selected portions associated with each of the checkpoints have not been tampered
15 with based on the protection tools used to produce the protected selected portions;
16 and

17 insert the plurality of checkpoints within the protected digital good,
18 the positions of each of the plurality of checkpoints being one of variably offset
19 and randomly offset outside of the protected selected portions with which each of
20 the checkpoints is associated.

1 **24.** (Original) A production system as recited in claim 23, wherein the
2 protection tools are selected from a group of protection tools comprising code
3 integrity verification, acyclic code integrity verification, cyclic code integrity
4 verification, secret key scattering, obfuscated function execution,
5 encryption/decryption, probabilistic checking, Boolean check obfuscation, in-
6 lining, reseeding pseudo random number generators with time varying inputs, anti-
7 disassembly methods, varying execution paths between runs, anti-debugging
8 methods, and time/space separation between tamper detection and response.

9
10 **25.** (Original) A production system as recited in claim 23, wherein the
11 production server applies a protection tool that enables a checksum to be
12 computed on a set of bytes of the digital good without actually reading the bytes.

13
14 **26.** (Original) A production system as recited in claim 23, wherein the
15 production server has a pseudo random generator to introduce randomness into the
16 application of the protection tools to various portions of the original digital good.
17
18
19
20
21
22
23
24
25

1 27. (Currently Amended) An obfuscation system, comprising:
2 a parser to parse a digital good into a plurality of segments;
3 a set of protection tools that may be applied to the segments of the digital
4 good to augment the segments with protection qualities;
5 a target segment selector to select at least one segment from the plurality of
6 segments;
7 a tool selector to select at least one protection tool from the set of protection
8 tools and apply the selected protection tool to the selected segment so that a
9 protection tool of the set of protection tools is applied only to a selected segment
10 of the plurality of segments to generate a plurality of protected selected segments;
11 and
12 a checkpoint generator to create checkpoints for at least a portion of the
13 protected selected segments, the checkpoints being assigned positions outside of
14 the protected selected segments at variable positions relative to each of the
15 protected selected segments, the checkpoints being operable to cause a system
16 receiving the plurality of protected selected segments, upon encountering the
17 checkpoints, to invoke a function call to attempt to validate authenticity of the
18 protected selected segments based on the protection tool applied to generate the
19 plurality of protected selected segments.
20
21
22
23
24
25

1 **28.** (Original) An obfuscation system as recited in claim 27, wherein the
2 protection tools are selected from a group of protection tools comprising code
3 integrity verification, acyclic code integrity verification, cyclic code integrity
4 verification, secret key scattering, obfuscated function execution,
5 encryption/decryption, probabilistic checking, Boolean check obfuscation, in-
6 lining, reseeding pseudo random number generators with time varying inputs, anti-
7 disassembly methods, varying execution paths between runs, anti-debugging
8 methods, and time/space separation between tamper detection and response.

9
10 **29.** (Original) An obfuscation system as recited in claim 27, wherein the target
11 segment selector comprises a pseudo random generator to enable random selection
12 of the segment.

13
14 **30.** (Original) An obfuscation system as recited in claim 27, wherein the tool
15 selector comprises a pseudo random generator to enable random selection of the
16 protection tool.

17
18 **31.** (Original) An obfuscation system as recited in claim 27, further
19 comprising a quantitative unit to specify a quantity of protection qualities to be
20 added to the digital good.

1 **32.** (Currently Amended) A client-server system, comprising:

2 a production server to randomly apply various forms of protection only to
3 selected portions of a digital good to produce a protected digital good, the
4 protected digital good including a plurality of one of variably and randomly placed
5 checkpoints configured to cause a system encountering the checkpoints ~~to attempt~~
6 to authenticate that the selected portions of the protected digital good have not
7 been tampered with; and

8 a client to store and execute the protected digital good, the client being
9 configured to, upon encountering each of the checkpoints, to invoke a function call
10 to evaluate the selected portions of the protected digital good to determine whether
11 the protected digital good has been tampered with based on at least one form of
12 protection applied to the selected portions to produce the protected digital good.

1 33. (Currently Amended) One or more computer-readable media having
2 computer-executable instructions that, when executed, direct a computing device
3 to:

4 parse a digital good into a plurality of segments;

5 apply multiple different protection tools to only a selected portion of the
6 segments in a random manner to produce a protected digital good having a
7 composite of variously protected portions; and

8 insert a plurality of checkpoints into the protected digital good at positions
9 one of variably and randomly offset from the variously protected portions, such
10 that upon encountering each of the plurality of checkpoints, a receiving computing
11 system executing attempting to execute the protected digital good will invoke a
12 function call to authenticate that variously protected portions have not been
13 tampered with based on the at least one of the multiple different protection tools
14 used to produce the variously protected portions.

15
16 34. (Original) One or more computer-readable media as recited in claim 33,
17 further comprising computer-executable instructions to randomly select the
18 protection tools from a set of available protection tools.

19
20 35. (Original) One or more computer-readable media as recited in claim 33,
21 further comprising computer-executable instructions to apply the protection tools
22 to randomly selected portions of the original digital good.

1 36. (Original) One or more computer-readable media as recited in claim 33,
2 wherein the protection tools are selected from a group of protection tools
3 comprising code integrity verification, acyclic code integrity verification, cyclic
4 code integrity verification, secret key scattering, obfuscated function execution,
5 encryption/decryption, probabilistic checking, Boolean check obfuscation, in-
6 lining, reseeding pseudo random number generators with time varying inputs, anti-
7 disassembly methods, varying execution paths between runs, anti-debugging
8 methods, and time/space separation between tamper detection and response.
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25